Package: OECDdata (via r-universe)

November 1, 2026

Type Package

Title Fast and Friendly Access to OECD Data

Version 0.3.4

Description Discover, describe, compare and import statistical datasets from the Organisation for Economic Co-operation and Development (OECD) via its public Statistical Data and Metadata eXchange (SDMX) Representational State Transfer (REST) API <https://sdmx.oecd.org/public/>. Adapts the discovery workflow of 'eurodata' to OECD-native names, catalogue-resolved short codes and dataset-scoped metadata. Uses SDMX REST version 2 by default, streamed downloads, native 'data.table' parsing, defensive bulk C XML extraction, indexed structural metadata resolution, caching and bounded HTTP retries. Includes an explicit version 1 transport for interoperability.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports curl (>= 5.0.0), data.table (>= 1.14.8), digest (>= 0.6.31), magrittr (>= 2.0.3), utils, xml2 (>= 1.3.3)

Suggests knitr, rmarkdown, testthat (>= 3.2.0), withr (>= 2.5.0)

VignetteBuilder knitr, rmarkdown

Config/testthat/edition 3

URL https://github.com/alekrutkowski/OECDdata

Config/checktor/acronyms OECD, SDMX, REST

SystemRequirements libxml2 (>= 2.9.0)

NeedsCompilation yes

Author Alek Rutkowski [aut, cre, cph]

Maintainer Alek Rutkowski <alek.rutkowski@gmail.com>

Config/pak/sysreqs libxml2-dev libssl-dev

Repository https://alekrutkowski.r-universe.dev

Date/Publication 2026-11-01 10:16:01 UTC

RemoteUrl https://github.com/alekrutkowski/oecddata

RemoteRef HEAD

RemoteSha 8d3473cd5916efd3c3624ded230affc87635a276

Contents

OECDdata-package . . . 2
as.OECDDataList . . . 3
browseDataList . . . 4
compare . . . 5
describe . . . 6
find . . . 7
importData . . . 8
importDataList . . . 9
importLabels . . . 10
importMetabase . . . 12
oecdDataset . . . 13
oecdExample . . . 14
oecdOptions . . . 15
oecdQuery . . . 16
oecdStructure . . . 17
print.OECDDataList . . . 18

Index **20**

`OECDdata-package` *Fast and Friendly Access to OECD Data*

Description

Discover, describe, compare and import OECD statistical datasets.

Details

The discovery and import workflow is inspired by **eurodata** 1.8.0 and adapted to OECD dataset identities, dimension dictionaries and observation attributes. Public arguments, result classes and metadata attributes use OECD terminology. Short codes are resolved from the catalogue; ambiguous matches are errors.

SDMX REST v2 is the default route. Observation payloads use SDMX-CSV 2 and structure payloads use SDMX-ML 2.1. The REST version and message version are independent. Version 1 routes are available by explicit option.

XML metadata use bulk C extraction through libxml2, with document-level indexes for structural definitions. Observation CSV files use **data.table**. Source installation requires a C compiler and libxml2 development files.

The package makes no requests and changes no options when loaded. Caching is session-local unless a directory is explicitly configured. Network examples are not run by ordinary checks. See the README for a worked tutorial, the installed vignette for offline examples, and `inst/ATTRIBUTION.md` in the source tree for provenance and dependency acknowledgments.

Value

The package provides functions for discovery, metadata inspection and flat data imports.

See Also

importData, importDataList, oecdDataset, oecdOptions

Examples

```
x <- oecdExample()
head(x)
stopifnot(is.numeric(x$value_))
```

---

as.OECDDataList *Coerce a Data Frame to an OECD Catalogue*

Description

Attach the OECD catalogue class and a descriptive search caption.

Usage

```
as.OECDDataList(x, SearchCriteria = "", ...)
```

Arguments

`x`	A data frame containing `Code`, `Dataset name` and `Link`.
`SearchCriteria`	A single descriptive string attached to the report.
`...`	Reserved for future extensions; currently unused.

Details

Other columns are preserved. No network access occurs.

Value

An independent data frame with classes `OECDDataList` and `data.frame`.

See Also

importDataList

Examples

```
x <- data.frame(Code = "TEST,DEMO,1.0", `Dataset name` = "Synthetic demo",
                Link = "https://example.invalid/", check.names = FALSE)
y <- as.OECDDataList(x, SearchCriteria = "offline example")
inherits(y, "OECDDataList")
```

`browseDataList` *Browse a Catalogue Subset*

Description

Browse a Catalogue Subset.

Usage

```
browseDataList(subs)
```

Arguments

`subs`	An R expression evaluating to a logical scalar or a logical vector as long as the catalogue. Column names are available directly; use backticks for names containing spaces. Caller variables are also available. Available columns are listed under Details. Missing values in the result are treated as false.

Details

The predicate in `subs` can refer to these catalogue columns:

`ShortCode` Unique, catalogue-resolved shorthand; missing for ambiguous tokens.

`Dataset name` Dataset title; refer to it with backticks.

`Code` Fully qualified agency, dataflow, version reference.

`Agency` Maintaining agency identifier.

`Dataflow` Complete dataflow identifier.

`DataflowPrefix` Identifier namespace before `@`.

`Version` Selected current structure version.

`Type` Artefact type (normally `dataset`).

`Last update of data` Source annotation for a data update, if present.

`Last table structure change` Source annotation for a structure update.

`Data start` Earliest period in source metadata, if reported.

`Data end` Latest period in source metadata, if reported.

`Description` Original OECD dataset description, which may contain HTML.

`Link` URL to the dataset in OECD Data Explorer.

The HTML report escapes ordinary catalogue fields. The Description column retains only safe formatting elements (such as paragraphs, lists and emphasis) and HTTP(S) or mailto links; scripts, event handlers and other potentially unsafe markup are excluded. Link URLs include OECD Data Explorer's data-space selector. Browser opening is a print side effect restricted to interactive sessions and controlled by `open_browser`. Assignment does not launch a browser. The expression is ordinary R code and should not be accepted from untrusted users.

Value

A list with `criteria`, `time`, `df`, and `html`, with class `BrowsedOECDDatasetList`.

See Also

find, importDataList

Examples

```
if (interactive()) {
browseDataList(grepl("employment", `Dataset name`, ignore.case = TRUE))
pattern <- "GDP|gross domestic"
browseDataList(grepl(pattern, `Dataset name`, ignore.case = TRUE))
}
```

---

compare *Compare Marginal Dimension-Code Coverage*

Description

Compare Marginal Dimension-Code Coverage.

Usage

```
compare(..., import_labels = TRUE, import_dim_labels = TRUE)
```

Arguments

`...`	Two or more different dataset references, each supplied as a separate scalar string.
`import_labels`	Logical; include code labels.
`import_dim_labels`	Logical; include dimension labels.

Details

Rows are the full union of dimension-ID/code pairs. Matching IDs are not proof of conceptual equivalence across datasets. Conflicting labels are joined with `" | "` and generate a warning rather than being silently overwritten. Availability follows importMetabase. This function compares metadata coverage, not numeric observations, units or revisions.

Value

A `data.table` with `Dim_name`, `Dim_val`, optional labels, and one logical column per input dataset. Missing membership is `FALSE`.

See Also

describe, importMetabase

Examples

```
if (interactive()) {
# Discover two relevant references first rather than guessing identifiers.
candidates <- OECDdata::find("national accounts")$df$Code
if (length(candidates) >= 2L) compare(candidates[[1]], candidates[[2]])
}
```

`describe` *Describe Dimensions and Marginal Code Coverage*

Description

Describe Dimensions and Marginal Code Coverage.

Usage

```
describe(OECDDatasetCode, import_labels = !wide, wide = FALSE,
         import_dim_labels = TRUE)
```

Arguments

`OECDDatasetCode`	A catalogue-resolved short code such as `"CLI"`, a terminal ID such as `"DF_CLI"`, a complete dataflow ID such as `"DSD_STES@DF_CLI"`, or a qualified reference `"OECD.SDD.STES,DSD_STES@DF_CLI"` with an optional numeric version. Short codes are accepted only when resolution is unambiguous. An omitted version, `"latest"` or `"+"` selects the latest stable structure. See oecdDataset for resolution rules.
`import_labels`	Logical; include human-readable code labels. Defaults to the opposite of `wide`.
`wide`	Logical; collapse codes into one comma-separated string per dimension.
`import_dim_labels`	Logical; include descriptive dimension names.

Details

The coverage semantics and limitations are those of importMetabase. `wide = TRUE` requires `import_labels = FALSE`. Label and shape defaults follow the familiar **eurodata** workflow. Results are returned normally; assignment does not force an extra print side effect.

Value

A `data.table`. Long form contains `Dim_name`, `Dim_val`, optional `Dim_name_label` and `Dim_val_label`, and a logical membership column named exactly as the input dataset reference. Wide form uses `Dim_values` instead of `Dim_val`.

See Also

compare, importMetabase

Examples

```
if (interactive()) {
code <- "CLI"
describe(code)
describe(code, wide = TRUE)
}
```

`find` *Search Dataset Codes or Titles*

Description

Search Dataset Codes or Titles.

Usage

```
find(...)
```

Arguments

`...`	Unquoted words, scalar numbers, or quoted regular-expression patterns. Prefix a term with unary minus for exclusion. Terms are captured rather than evaluated as function calls.

Details

Matching is case-insensitive and uses partial regular expressions. As in **eurodata**, all positive and negative conditions must hold in Code, OR all must hold in Dataset name. Terms are not matched across a concatenation of both fields. Consequently an exclusion in the title does not veto a successful Code match. Use browseDataList for explicitly global exclusions.

Use `OECDdata::find` to avoid confusion with `utils::find`. Printing shows a text report, using a text viewer only in interactive mode when enabled. Unlike eurodata's search report, the returned data frame retains the full catalogue schema, including missing optional columns.

Value

A list with `criteria`, `time`, `df`, and `report`, with class `FoundOECDDatasetList`.

See Also

browseDataList, importDataList

Examples

```
if (interactive()) {
OECDdata::find(gdp, annual, -quarterly)
OECDdata::find("health expenditure")
OECDdata::find("climat|emission")
}
```

`importData` *Import a Flat OECD Dataset*

Description

Import a Flat OECD Dataset.

Usage

```
importData(OECDDatasetCode, filters = NULL)
```

Arguments

`OECDDatasetCode`	A catalogue-resolved short code such as `"CLI"`, a terminal ID such as `"DF_CLI"`, a complete dataflow ID such as `"DSD_STES@DF_CLI"`, or a qualified reference `"OECD.SDD.STES,DSD_STES@DF_CLI"` with an optional numeric version. Short codes are accepted only when resolution is unambiguous. An omitted version, `"latest"` or `"+"` selects the latest stable structure. See oecdDataset for resolution rules.
`filters`	`NULL` for all observations, or a non-empty named list of character/numeric vectors. Names identify dimensions, except for the optional reserved `.dataset` block described below. Multiple codes within a dimension are combined with OR; dimensions are combined with AND. A time vector is interpreted as an inclusive range from its minimum to its maximum, not as a discrete set of requested dates. See Details.

Details

The optional `.dataset` block is a named list of `agency`, `prefix` and/or `version` identity selectors, with the same meaning as in oecdDataset. For example, `list(.dataset = list(agency = "OECD.SDD.STES"), REF_AREA = "USA")`. These selectors choose a dataset and are removed before dimension filtering. A selector-only block leaves all dimensions unrestricted. Conflicts with an explicit reference are errors. The `Filters` result attribute retains the original full list, including the selector block.

The structure is resolved before constructing the data query. Dimension names are matched case-insensitively; values must be valid literal codes with their original case. Leading zeros in identifiers are preserved. A single empty string selects all values of its dimension and produces a warning. Other missing or empty selections fail explicitly.

Common annual, quarterly, half-yearly, weekly, monthly and daily period formats are supported: `YYYY`, `YYYY-Q1`, `YYYY-S1`, `YYYY-W01`, `YYYY-MM`, and `YYYY-MM-DD`. Annual `YYYY-A1` and monthly `YYYY-M01` aliases are normalized. Use one granularity in a filter. Subdaily periods, custom calendars and mixed-granularity bounds are not supported.

The client downloads to disk and parses with `data.table::fread`. It does not aggregate, infer seasonal adjustment, rescale units, translate flags, or fill in a Cartesian product of missing observations. Duplicate observation keys, delete actions, unexpected response identities and malformed values fail rather than being silently repaired. HTTP 204 produces a typed empty result with a warning; HTTP errors, including 404, remain errors.

A pinned structure version does not freeze historical statistical revisions. Save an RDS snapshot when an exact data vintage is required.

Value

By default, a `data.frame` with classes `OECDDataset` and `data.frame`. Dimension columns are factors, `value_` is a double, and `flags_` contains the raw `OBS_STATUS` text, or an empty string if that attribute is absent. All other supplied statistical attributes are retained. Set `return_data_table = TRUE` for a `data.table`, and `strings_as_factors = FALSE` for character dimensions.

Attributes include `RequestedDatasetCode` (input reference), `OECDDatasetCode` (resolved exact reference), `DownloadTime`, `SourceURL`, `Cached`, `Dimensions`, `OECDAttributes`, and `Filters`. A cache hit preserves the original download time.

See Also

oecdDataset, oecdQuery, oecdStructure, importLabels, oecdOptions

Examples

```
x <- oecdExample()
names(x)
attr(x, "Dimensions")
if (interactive()) {
cli <- importData(OECDDatasetCode = "CLI",
  filters = list(REF_AREA = c("USA", "FRA"), FREQ = "M",
                 TIME_PERIOD = c("2023-01", "2023-12")))
}
```

`importDataList` *Import the Current OECD Dataflow Catalogue*

Description

Retrieve dataset titles, readable codes and canonical identities.

Usage

```
importDataList()
```

Details

There is one current structure version per agency/dataflow identity. `Code` is the fully qualified reference with that version. `ShortCode` is a compact, catalogue-resolved shorthand, or missing when another identity could match the same token. It is a package convenience, not an official global OECD alias. `DataflowPrefix` is the namespace before `@`, or missing when that delimiter is absent. See oecdDataset for matching rules.

No artificial hierarchy columns are supplied. Timestamps and period bounds are populated only when recognized source annotations supply them; other values remain missing. A retrieval time is not a last-data-update time.

The catalogue belongs to the configured `base_url`. Its printable HTML report escapes ordinary cells and renders a restricted, safe subset of HTML from OECD descriptions. Dataset links include the OECD Data Explorer space selector `df[ds]=dsDisseminateFinalDMZ`. Printing opens a browser only in an interactive session with `open_browser = TRUE`.

Value

A data frame with classes `OECDDataList` and `data.frame`. Columns are `ShortCode`, `Dataset name`, `Code`, `Agency`, `Dataflow`, `DataflowPrefix`, `Version`, `Type`, `Last update of data`, `Last table structure change`, `Data start`, `Data end`, `Description`, and `Link`.

See Also

find, browseDataList, as.OECDDataList, oecdDataset

Examples

```
if (interactive()) {
catalogue <- importDataList()
head(catalogue[c("ShortCode", "Dataset name", "Code")])
}
```

---

importLabels *Import Dataset-Scoped Dimension or Attribute Labels*

Description

Import Dataset-Scoped Dimension or Attribute Labels.

Usage

```
importLabels(OECDDimCode, dataset = NULL)

importDimLabel(OECDDimCode, dataset = NULL)
```

Arguments

OECDDimCode	A component ID such as "REF_AREA". Without dataset context, a fully qualified codelist reference or a globally unambiguous CL_-prefixed codelist can be used. A single string of the form "AGENCY,DATAFLOW,VERSION::DIMENSION" also supplies context.
dataset	Optional dataset short code or canonical reference, as in oecdDataset. Defaults to the dataset option. Explicit dataset context is recommended because OECD codelist IDs need not be globally unique. A qualified component reference overrides the default option; an explicitly supplied dataset must resolve to the same dataset.

Details

Labels are resolved from the specific DSD, concept scheme and enumerated codelist, including agency and version. The requested language is preferred, then English, then the first available label. Ambiguous dictionaries cause an error rather than an arbitrary selection. Labels are permissible codes, not a claim that observations exist. A component without an enumerable dictionary can be passed to importDimLabel with dataset context to retrieve its name, but cannot be passed to importLabels. Use oecdDataset first when agency, prefix or version selectors are needed. Attributes identify the OECDDimCode, resolved OECDDatasetCode, and DownloadTime.

Value

importLabels returns a two-column data frame with classes OECDLabels and data.frame. The first column uses the component name supplied by the caller; the second appends _labels. Columns are factors by default. importDimLabel returns one character label.

See Also

oecdStructure, describe

Examples

```
if (interactive()) {
code <- "CLI"
importLabels("REF_AREA", dataset = code)
importDimLabel("REF_AREA", dataset = code)
old <- oecdOptions(dataset = code)
importLabels("FREQ")
options(old)
}
```

`importMetabase` *Import a Dataset-Dimension-Code Inventory*

Description

Import a Dataset-Dimension-Code Inventory.

Usage

```
importMetabase(OECDDatasetCode = NULL)
```

Arguments

`OECDDatasetCode`
: Optional character vector of dataset short codes or canonical references. See oecdDataset for resolution rules. Defaults to the `dataset` option. With neither supplied, attempts the complete catalogue inventory, which can be very large.

Details

By default, actual marginal availability must be enumerated by a matching SDMX Actual content constraint. A marginal list does not guarantee observations for every combination of the listed codes. Missing metadata or constraints whose conditional exclusions cannot safely be projected cause an explicit error.

Set `oecdOptions(availability = "allowed")` to request dictionary codes instead. This is an explicit alternate mode, not a claim of observed coverage. The source may return partial dictionaries. Prefer dataset-scoped requests over a global inventory. The no-argument call is retained for compatibility but is expensive and may fail if any selected dataset lacks usable metadata.

Value

A data frame with columns `Code`, `Dim_name`, and `Dim_val`. The `Availability` attribute records `"actual"` or `"allowed"`. Time dimensions are excluded.

See Also

describe, compare, oecdStructure

Examples

```
if (interactive()) {
importMetabase("CLI")
}
```

`oecdDataset` *Resolve an OECD Dataset to a Canonical Reference*

Description

Resolve a readable dataset code, optional identity selectors and its structure version without downloading observations.

Usage

```
oecdDataset(OECDDatasetCode, agency = NULL, prefix = NULL, version = NULL)
```

Arguments

`OECDDatasetCode`	A short code, a complete dataflow ID, or a reference `"AGENCY,DATAFLOW[,VERSION]"`. For example `"CLI"`, `"DF_CLI"`, `"DSD_STES@DF_CLI"`, or `"OECD.SDD.STES,DSD_STES@DF_CLI"`.
`agency`	Optional single agency identifier, for example `"OECD.SDD.STES"`.
`prefix`	Optional dataflow namespace prefix before `@`, for example `"DSD_STES"`. This selects the prefix of the dataflow identifier; the actual referenced data structure is still resolved from the returned metadata.
`version`	Optional numeric version string, `"latest"`, or `"+"`. An omitted version selects the catalogue's current stable version. An explicit older version is requested even when the current catalogue lists a newer one; the service must still supply metadata for that older version.

Details

This is a package-level convenience, not a different OECD API. For unqualified input, matching is case-insensitive and follows three levels: complete dataflow ID, terminal ID after the last `@`, then that terminal ID with a leading `DF_` removed. The first non-empty level is used. No fuzzy matching, legacy-code translation or hard-coded alias table is used.

Selectors restrict the catalogue before matching. If the chosen level contains more than one agency/dataflow identity, an `OECDdata_ambiguous` error lists canonical candidates and carries their complete rows in `candidates`. Use the agency or prefix selector, or a canonical reference. Selectors that conflict with an explicitly qualified reference are errors, not overrides.

importDataList advertises a `ShortCode` only if it cannot collide with any other identity's literal, terminal or compact code. Otherwise that column is missing. A short code can become ambiguous after catalogue changes; store this function's canonical result for repeatable dataset selection.

A qualified reference bypasses the catalogue, but its structure is still fetched or read from a verified cache. Offline shorthand resolution needs a cached catalogue in addition to cached structure/data responses. This function fetches structure metadata and dictionaries, but never observations. Returned structure versions do not freeze future observation revisions.

importData and oecdQuery also accept these selectors in a reserved `filters = list(.dataset = list(agency = ..., prefix = ..., version = ...),REF_AREA = ...)` block. Dataset selectors

are removed before building observation filters. A block containing only dataset selectors leaves all dimensions unrestricted. Metadata functions accept this function's returned reference.

Value

One character string containing the canonical, fully qualified "AGENCY,DATAFLOW,VERSION" reference with an exact resolved version.

See Also

importData, importDataList, oecdQuery, oecdStructure

Examples

```
if (interactive()) {
ref <- oecdDataset("CLI", agency = "OECD.SDD.STES", prefix = "DSD_STES")
s <- oecdStructure(ref)
importLabels(OECDDimCode = "REF_AREA", dataset = ref)
q <- oecdQuery("CLI", filters = list(
  .dataset = list(agency = "OECD.SDD.STES"),
  REF_AREA = "USA", FREQ = "M", TIME_PERIOD = c("2023-01", "2023-12")))
attr(q, "OECDDatasetCode")
}
```

---

oecdExample *Load a Small Synthetic Dataset Without Network Access*

Description

Load a Small Synthetic Dataset Without Network Access.

Usage

```
oecdExample()
```

Details

The bundled structure and CSV are parsed through the same core code paths as live responses. The synthetic reference is TEST,DEMO,1.0; its illustrative metadata timestamp is fixed. This function makes no requests and creates no cache files. Current output-type options are respected.

Value

Twelve synthetic monthly observations for two country codes, in the same format as importData. Values, structures and labels are invented teaching fixtures, not OECD statistics.

See Also

importData

Examples

```
x <- oecdExample()
head(x)
stopifnot(nrow(x) == 12L, is.numeric(x$value_))
attr(x, "OECDDatasetCode")
```

`oecdOptions` *Read or Set OECDdata Options*

Description

Read or Set OECDdata Options.

Usage

```
oecdOptions(...)

clearCache()
```

Arguments

`...`	Named option values without the `OECDdata.` prefix. With no arguments, return all current settings.

Details

All settings are ordinary R options prefixed with `OECDdata.`.

api_version `"v2"` by default; explicitly select `"v1"` for legacy routes.

base_url `"https://sdmx.oecd.org/public/rest"`. An HTTPS root ending in `/rest`, without `/v2`.

language `"en"`; requested metadata language.

dataset `NULL`; default context for labels and scoped metabase calls.

cache, cache_dir `TRUE`, `NULL`; a session temporary directory unless explicitly configured.

cache_ttl, metadata_ttl `3600`, `86400` seconds, respectively.

offline, refresh Both `FALSE`. Offline accepts expired but verified entries. Refresh bypasses cache reuse online.

timeout, connect_timeout `120`, `20` seconds.

retries, max_retry_wait `3`, `60`; retry count after the initial attempt and maximum allowed retry delay in seconds.

min_interval `1` second between requests in this session.

max_downloads, download_window `60`, `3600`; local rolling data-attempt budget. Retries count; metadata requests do not.

strings_as_factors, return_data_table `TRUE`, `FALSE`; compatible default result types.

quiet, open_browser `FALSE`, `interactive()`; suppress progress messages or control interactive reports.

availability `"actual"`; use `"allowed"` only for explicitly requested dictionary-code inventories.

max_bytes `Inf`; optional response-size limit in bytes.

HTTP Retry-After is respected. Delays above the configured maximum produce an error, never an early retry. A local budget cannot account for other sessions sharing an IP address. It is not a guarantee against service throttling.

Cache identities include URL, media type and language. Entries carry checksums, status and original download times. Only package-owned files are removed by `clearCache`. No shared-process locking or automatic eviction is provided; use a separate cache directory per concurrent writer. Checksums detect corruption but are not a substitute for trusting the configured local directory. The setter returns the prior namespaced settings invisibly. In a function, use `old <- oecdOptions(...)` and `on.exit(options(old), add = TRUE)` to restore them when leaving.

The size limit is enforced by libcurl when the server advertises a usable length and is checked after transfer and before returning a cached response as well. It is not an absolute bound on bytes received when a response has no advertised size.

Value

With no arguments, `oecdOptions` returns a named settings list. When setting options, it invisibly returns the previous, fully prefixed options, so `options(old)` restores them. `clearCache` invisibly returns the number of owned cache files removed, not the number of requests.

See Also

importData, importMetabase

Examples

```
old <- oecdOptions(quiet = TRUE, strings_as_factors = FALSE)
x <- oecdExample()
stopifnot(is.character(x$REF_AREA))
options(old)
```

---

oecdQuery *Construct an Encoded OECD Data Request*

Description

Construct an Encoded OECD Data Request.

Usage

```
oecdQuery(OECDDatasetCode, filters = NULL)
```

Arguments

OECDDatasetCode	A catalogue-resolved short code such as "CLI", a terminal ID such as "DF_CLI", a complete dataflow ID such as "DSD_STES@DF_CLI", or a qualified reference "OECD.SDD.STES,DSD_STES@DF_CLI" with an optional numeric version. Short codes are accepted only when resolution is unambiguous. An omitted version, "latest" or "+" selects the latest stable structure. See oecdDataset for resolution rules.
filters	Same named filter list as importData, including its optional reserved .dataset selector block.

Details

This can fetch structural metadata but never downloads observations. V2 uses named component constraints with encoded comma-OR and plus-AND operators. V1 uses positionally ordered keys and start/end period parameters. Reserved literal filter characters are rejected. The URL alone does not include the HTTP Accept header; send the attached Accept value when inspecting it with another client.

Value

A character URL with attributes Accept, RequestedDatasetCode and OECDDatasetCode. The URL contains an exact resolved structure version.

See Also

importData, oecdOptions

Examples

```
if (interactive()) {
q <- oecdQuery("CLI",
               list(REF_AREA = "USA", TIME_PERIOD = c("2023-01", "2023-12")))
cat(q)
attr(q, "Accept")
}
```

---

oecdStructure *Inspect the Resolved OECD Structure*

Description

Inspect the Resolved OECD Structure.

Usage

```
oecdStructure(OECDDatasetCode)
```

Arguments

`OECDDatasetCode`

A catalogue-resolved short code such as `"CLI"`, a terminal ID such as `"DF_CLI"`, a complete dataflow ID such as `"DSD_STES@DF_CLI"`, or a qualified reference `"OECD.SDD.STES,DSD_STES@DF_CLI"` with an optional numeric version. Short codes are accepted only when resolution is unambiguous. An omitted version, `"latest"` or `"+"` selects the latest stable structure. See oecdDataset for resolution rules.

Details

This makes a metadata request, not an observation download. The referenced DSD and its dictionaries are resolved by full identity, not XML document order. Only structures with a single observation measure are supported. The dictionary and observed marginal availability are separate elements. Inspect the latter's status and explanation before interpreting it as coverage.

Value

An `OECDStructure` list with `flow`, `dsd`, `dimensions`, `attributes`, `measure`, `labels`, `available`, `availability`, and `availability_reason`. The flow includes its resolved agency, identifier, version, title, description and canonical code. The `dsd` entry records the resolved DSD agency, identifier and version. Component tables include ID, label, kind, position and codelist reference. Download time and source URL are attributes.

See Also

oecdQuery, importData

Examples

```
if (interactive()) {
s <- oecdStructure("CLI")
s$flow
s$dimensions
s$labels$REF_AREA
s$availability
}
```

`print.OECDDataList` *Print OECD Search and Catalogue Reports*

Description

Print OECD Search and Catalogue Reports.

Usage

```
## S3 method for class 'OECDDataList'
print(x, SearchCriteria = attr(x, "SearchCriteria"), ...)
## S3 method for class 'FoundOECDDatasetList'
print(x, ...)
## S3 method for class 'BrowsedOECDDatasetList'
print(x, ...)
## S3 method for class 'FoundOECDDatasetListReport'
print(x, ...)
```

Arguments

x	An object returned by the corresponding catalogue or search function.
SearchCriteria	Optional catalogue report caption.
...	Further print arguments; used by the data-frame console report.

Details

Interactive browser or text-viewer reports are enabled only when the option open_browser is true and R is interactive. Otherwise reports are printed to the console. No browser is launched by ordinary package checks.

Value

The input, invisibly.

See Also

find, browseDataList, oecdOptions

Examples

```
x <- as.OECDDataList(data.frame(
  Code = "TEST,DEMO,1.0",
  `Dataset name` = "Synthetic demonstration",
  Link = "https://example.org/",
  check.names = FALSE
))
print(x)
```

Index

as.OECDDataList, 3, *10*

browseDataList, 4, *7, 8*, *10*, *19*

clearCache *(*oecdOptions*)*, 15
compare, 5, *7*, *12*

describe, 6, *6*, *11, 12*

find, *5*, 7, *10*, *19*

importData, *3*, 8, *13, 14*, *16–18*
importDataList, *3*, *5*, *8*, 9, *13, 14*
importDimLabel *(*importLabels*)*, 10
importLabels, *9*, 10
importMetabase, *5–7*, 12, *16*

OECDdata *(*OECDdata-package*)*, 2
OECDdata-package, 2
oecdDataset, *3*, *6*, *8–12*, 13, *17, 18*
oecdExample, 14
oecdOptions, *3*, *9*, 15, *17*, *19*
oecdQuery, *9*, *13, 14*, 16, *18*
oecdStructure, *9*, *11, 12*, *14*, 17

print.BrowsedOECDDatasetList
        *(*print.OECDDataList*)*, 18
print.FoundOECDDatasetList
        *(*print.OECDDataList*)*, 18
print.FoundOECDDatasetListReport
        *(*print.OECDDataList*)*, 18
print.OECDDataList, 18